REMARKS

Claims 1-21 were presented for examination and were pending in this application. In an Office Action dated January 12, 2007 claims 1-21 were rejected. With this amendment, Application has filed a request for continued examination. Applicant has added new claims 22-27. Applicant has amended claims 1, 6 and 14 and now respectfully requests consideration of the application in view of the above-Amendment and the following Remarks.

Claims 1-21 were rejected under 35 USC §102(e) as allegedly being anticipated by U.S. Patent No. 6,173,316 to DeBoor ("DeBoor").

Statement of Substance of Interview

On January 25, 2007, Applicant's representatives Greg Sueoka and Brian Brannon conducted an interview with examiner Andre R. Fowlkes. In this interview, the pending rejections of all claims were discussed.

During the interview, Applicant's representatives noted the issuance of a European patent from a foreign application corresponding to this application. Further, Applicant's representatives argued that the cited DeBoor reference does not disclose the claimed feature of "integrating executable instructions within at least one from a group of an XML-document and a document type definition (DTD)." Specifically, Applicant's representatives pointed out that DeBoor merely recites a man-machine interface stored on a wireless device that uses a markup language to format data for display on the wireless device, but does not recite "integrating executable instructions," such as database queries or mathematical computations, into an XML-document or document type definition. Additionally, Applicant's representatives pointed out that DeBoor fails to disclose "storing intermediate states of the execution process in a memory of the data processing system by dynamically creating and redefining attributes of elements of the XML document," but discloses replacing HTML tags to control formatting, or store values in a form, rather than storing intermediate states of the execution process, such as temporary values used in computation or retrieved data, using attributes in the XML document. Examiner indicated these arguments distinguish the claimed invention from the cited reference, but a further search would be required. Accordingly, to advance prosecution of this application, Applicant has filed a request for continued examination and provided these arguments in writing.

Claim Objections

Examiner has objected to claim 6 because of informalities. Applicant has amended claim 6 accordingly. Thus, Applicant respectfully submits that all informalities with respect to claim 6 have been resolved.

Claim Rejections - 35 USC § 102

Claims 1-21 were rejected under 35 USC §102(e) as allegedly being anticipated by U.S. Patent No. 6,173,316 ("DeBoor"). This rejection is respectfully traversed.

Claim 1, as amended, recites:

> A method for the direct execution of an XML-document in a data processing system, comprising:
>
> defining a local behavior and process for each element of the XML-document;
>
> integrating executable instructions with at least one XML-document or a document type definition (DTD); and
>
> storing intermediate states of the execution process in a memory of the data processing system by dynamically creating and redefining attributes of elements of the XML document,

> where the intermediate states define intermediate states of an execution of the executable instructions.

Thus, amended claim 1 recites defining a local behavior and process for each element of an XML-document. Executable instructions are then integrated with at least one XML-document or a document type definition (DTD). By integrating executable instructions with an XML-document or DTD definition, the XML-document or DTD is used to execute a program, such as a mathematical computation, database query, data storage or other similar operations. Execution of the executable instructions causes intermediate states of the execution process to be stored in a memory by dynamically creating and redefining attributes of elements of the XML-document. Thus, as the program defined by the XML-document is executed, the XML-document itself is modified to reflect the program execution by creating or modifying elements of the XML-document according to states in the program execution.

In contrast, DeBoor discloses a man-machine-interface (MMI) in which "Internet and World Wide Web access features are seamlessly integrated with the telephony and other controls of the wireless communication device so that user can access any feature of the wireless communication device at any time." DeBoor, col. 2, lines 42-46. The MMI disclosed in DeBoor allows service operators to "quickly and easily brand the wireless communication device with their own distinctive user interfaces." DeBoor, col. 6, lines 15-17. Thus, the MMI disclosed in DeBoor resides on a wireless communication device, not in an XML-document or DTD, and merely specifies how content is displayed on the wireless communication device. As the disclosed MMI merely decodes and displays received content, DeBoor fails to disclose the "execution of executable instructions" as recited in claim 1.

Furthermore, DeBoor fails to disclose “storing intermediate states of the execution process in a memory of a data processing system by dynamically creating and redefining attributes of elements of the XML document, where the intermediate states define intermediate states of an execution of the executable instructions” as recited in claim 1. In contrast, DeBoor discloses "(intermediate states of the process are stored and) replacing the tag with the second markup language page to form a combined markup language page (i.e. the elements are redefined)." DeBoor, col. 9, lines 9-10. This discloses the operation of a conventional markup language used by web-publishing systems for rendering the content of a web page for display on a wireless device, rather than executing “executable instructions.” In describing the MMI, DeBoor explicitly states:

> Content handlers 114 are responsible for decoding the content data of a page corresponding to a fetched URL and displaying the content, or otherwise manipulating the content. A content handler 114 typically decodes content it receives and presents a page in the screen display 136, or portion thereof. Some content handlers 114 construct the page from data they receive from the memory 126 or over a communications link, while others display the state of the wireless communication device 100 or serve some other administrative function.

DeBoor, col. 12, lines 37-46. Thus, the MMI of DeBoor merely determines how data is presented to a user by interpreting tags of a markup language to format the data. Replacing tags with markup pages to form a combined markup language page does not involve “execution of the executable instructions,” but merely displays data according to a particular layout. Thus, in DeBoor, there are no intermediate states of an execution, but merely a combined markup language page generated from replacing tags. Hence, DeBoor fails to disclose “storing intermediate states of the execution process in a memory of a data processing system by dynamically creating and redefining attributes of elements of the XML document, where the intermediate states define intermediate states of an execution of the executable instructions," as recited in claim 1.

Thus, Applicant respectfully submits that for at least these reasons, claim 1 is patentably distinguishable over DeBoor. Therefore, Applicant respectfully requests that Examiner reconsider the rejection and withdraw it.

As claims 2-6 are dependent from claim 1, all arguments advanced above with respect to claim 1 are hereby incorporated so as to apply to claims 2-6. Thus, Applicant respectfully submits that for at least these reasons, claims 2-6 are patentably distinguishable over DeBoor. Therefore, Applicant respectfully requests that Examiner reconsider the rejection and withdraw it.

Amended claim 14 similarly recites "integrating executable instructions with at least one of a document type definition (DTD) or an XML-document" and "storing intermediate states by dynamically creating and redefining element attributes," so all arguments advanced above with respect to claim 1 are hereby incorporated so as to apply to claim 14. Thus, Applicant respectfully submits that for at least these reasons, claim 14 is patentably distinguishable over DeBoor. Therefore, Applicant respectfully requests that Examiner reconsider the rejection and withdraw it.

Claim 21 also recites similar limitations of "integrating executable instructions with at least one XML-document or a document type definition (DTD)" and "storing intermediate states of the execution process in a memory of the data processing system by dynamically creating and redefining elements." Hence, all arguments advanced above with respect to claim 1 are hereby incorporated so as to apply to claim 21. Thus, Applicant respectfully submits that for at least these reasons, claim 21 is patentably distinguishable over DeBoor.

Therefore, Applicant respectfully requests that Examiner reconsider the rejection and withdraw it.

Independent claim 7 recites:

> A system for use with the method according to one of the preceding claims, comprising:
>
> a server providing services to at least one client by executing at least parts of a XML-document according to a XML-robot specification sent from the client to the server or a server providing services to at least one client by sending a XML-robot specification and a XML-document to the client, such that said service is provided by executing at least part of the sent document on the client according to the sent XML-robot specification.

Thus, claim 7 provides that a server sends an XML-robot specification and a XML document to a client, where at least part of the XML-document is executed by the client according to the provided XML-robot specification or a server receives an XML-robot specification and an XML-document. This allows the server to provide services by transmitting to a client an XML-robot specification describing how to execute instructions provided in at least parts of an XML-document, or by the server executing instructions in at least part of an XML document according to a XML-robot specification provided by the client. Hence, the service is provided when the client or the server executes at least part of the XML-document according to the provided XML-robot specification.

DeBoor merely discloses a MMI which "parses the [received] content and displays it on a screen display." DeBoor, col. 4, lines 38-40. The MMI in DeBoor "is generally responsible for displaying any fetched markup language pages, including all user interface pages, and for receiving user inputs to these pages via forms and other input means." De Boor, col. 4, lines 40-44. DeBoor does not disclose "executing at least part of the sent document on the client according to the sent XML-robot specification," as recited in claim 7.

Rather, DeBoor discloses receiving content and formatting the content for display and fails to disclose "a XML-robot specification" which specifies how to execute at least part of a XML document.

Thus, Applicant respectfully submits that for at least these reasons, claim 7 is patentably distinguishable over DeBoor. Therefore, Applicant respectfully requests that Examiner reconsider the rejection and withdraw it.

As claims 8-10 similarly recite "XML-robot specification," all arguments advanced above with respect to claim 7 are hereby incorporated so as to apply to claims 8-10. Thus, Applicant respectfully submits that for at least these reasons, claims 8-10 are patentably distinguishable over DeBoor. Therefore, Applicant respectfully requests that Examiner reconsider the rejection and withdraw it.

Independent claim 11 recites:

> An apparatus for the direct execution of XML documents, comprising:
>
> means for graphical display of XML-robot specifications within an advanced visual integrated development environment; and
>
> means for generating animations of the execution process.

Thus, claim 11 provides an apparatus which graphically displays an XML-robot specification and generates animations of the execution process of directly executed XML documents.

In contrast, DeBoor discloses a man-machine-interface (MMI) in which "Internet and World Wide Web access features are seamlessly integrated with the telephony and other controls of the wireless communication device so that user can access any feature of the wireless communication device at any time." DeBoor, col. 2, lines 42-46. The MMI disclosed in DeBoor allows service operators to “quickly and easily brand the wireless communication device with their own distinctive user interfaces.” DeBoor, col. 6, lines 15-17. Thus, the MMI displays content on the wireless communication device, and does not execute the received content. As the MMI in DeBoor does not execute the received content, there is “no execution process,” and no “animations of the execution process,” as recited in claim 11.

Thus, Applicant respectfully submits that for at least these reasons, claim 11 is patentably distinguishable over DeBoor. Therefore, Applicant respectfully requests that Examiner reconsider the rejection and withdraw it.

Independent claim 12 recites:

> A method for the direct execution of XML documents comprising:
>
> providing an execution specification including
>
> a DTD;
>
> graphical flow charts; and
>
> transition rules;
>
> providing an XML document instance including
>
> an XML document;
>
> using the DTD to validate the XML document;
>
> constructing an attributed structure tree;
>
> decorating the attributed structure tree with the graphical flow charts to create a global flow chart; and
>
> executing the global flow chart according to the transition rules to directly execute the XML document.

Thus, claim 12 provides a method for the direct execution of XML documents by providing an execution specification including a DTD, graphical flow charts and transition rules. An XML document instance, including an XML document, is also provided. The DTD provided in the execution specification is then used to validate the XML document and construct an attributed structure tree. A global flow chart is then created by decorating the attributed structure tree with the graphical flow charts provided by the execution specification. The global flow chart is then executed according to the transition rules to directly execute the XML document.

In contrast, DeBoor discloses a man-machine-interface (MMI) in which "Internet and World Wide Web access features are seamlessly integrated with the telephony and other controls of the wireless communication device so that user can access any feature of the wireless communication device at any time." DeBoor, col. 2, lines 42-46. The MMI allows service operators to "quickly and easily brand the wireless communication device with their own distinctive user interfaces." DeBoor, col. 6, lines 15-17. Because the MMI merely parses and displays received content on a screen display, there is no "providing an execution specification" to the MMI but merely receiving of content to be displayed. *See* DeBoor, col. 4, lines 38-40. Thus, DeBoor fails to disclose claim 12.

Further, DeBoor fails to disclose that the execution specification includes "graphical flow charts." Other than describing the figures in the patent application, DeBoor does not disclose flow charts. The portion of DeBoor cited by Examiner appears in the "Brief Description of the Drawings" section and provides that "FIG. 6 is a flowchart of the operation of the HTMLp content handler in processing a string input associated with a user interface gadget." DeBoor, col. 9, lines 49-51. Figure 6 merely describes how the HTMLp content handler operates; and does not disclose a component of the data used by the MMI for displaying content. However, there is no indication that the flow chart of Figure 6 is included in an "execution specification," as recited in claim 12. Rather, the MMI in DeBoor receiving tags and uses the tags to present data for display, and there is no disclosure of the

MMI receiving flowcharts. Hence, DeBoor also fails to disclose "an execution specification including…graphical flow charts," as recited in claim 12.

Thus, Applicant respectfully submits that for at least these reasons, claim 12 is patentably distinguishable over DeBoor. Therefore, Applicant respectfully requests that Examiner reconsider the rejection and withdraw it

As claim 13 similarly recites "an execution specification including…graphical flow charts," the arguments advanced above with respect to claim 12 are hereby incorporated so as to apply to claim 13. Thus, Applicant respectfully submits that for at least these reasons, claim 13 is patentably distinguishable over DeBoor. Therefore, Applicant respectfully requests that Examiner reconsider the rejection and withdraw it.

Independent claim 15 recites:

> A system for the execution of an XML document comprising
>
> an interpreter generator having an input and an output, the input operative to receive an XML specification, the interpreter generator operative to produce at the output an interpreter, the interpreter having an input and an output, the input operative to receive an XML document, the interpreter operative to validate the XML document with respect to a general DTD and to execute the XML document.

Thus, claim 15 recites an interpreter generator which receives as input an XML specification and produces as output an interpreter. The generated interpreter then receives an XML document, validates the received XML document and executes the XML document. This allows an XML specification to generate an interpreter for directly executing additional XML documents.

In contrast, DeBoor discloses a man-machine-interface (MMI) in which "Internet and World Wide Web access features are seamlessly integrated with the telephony and other controls of the wireless communication device so that user can access any feature of the wireless communication device at any time." DeBoor, col. 2, lines 42-46. Further, the MMI disclosed in DeBoor allows service operators to "quickly and easily brand the wireless communication device with their own distinctive user interfaces." DeBoor, col. 6, lines 15-17. Thus, the MMI merely displays received content without the "execution of an XML document." Further, the MMI does not "produce at the output an interpreter," but merely produces a formatted display of the received content. No interpreter is generated as the MMI itself formats the display according to the received content. Thus, DeBoor fails to disclose "an interpreter generator having an input and an output, the input operative to receive an XML specification, the interpreter generator operative to produce at the output an interpreter" as recited in claim 15.

Thus, Applicant respectfully submits that for at least these reasons, claim 15 is patentably distinguishable over DeBoor. Therefore, Applicant respectfully requests that Examiner reconsider the rejection and withdraw it.

Independent claim 16 recites:

> A system for the execution of an XML document comprising:
>
> a compiler generator having an input and an output, the input operative to receive an XML specification, the compiler generator operative to produce at the output a compiler, the compiler having an input and an output, the input operative to receive a XML document valid with respect to a general DTD, the compiler operative to produce an executable document at the output.

Thus, claim 16 recites a compiler generator which receives as input an XML specification and produces as output a compiler. The generated compiler then receives an XML document, validates the received XML document and produces an executable document as the output. This allows an XML specification to generate a compiler for producing an executable document as output.

In contrast, DeBoor discloses a man-machine-interface (MMI) in which "Internet and World Wide Web access features are seamlessly integrated with the telephony and other controls of the wireless communication device so that user can access any feature of the wireless communication device at any time." DeBoor, col. 2, lines 42-46. Further, the MMI disclosed in DeBoor allows service operators to "quickly and easily brand the wireless communication device with their own distinctive user interfaces." DeBoor, col. 6, lines 15-17. Thus, the MMI merely displays received content, and does not "produce an executable document at the output." Further, the MMI does not "produce at the output a compiler," but merely produces a formatted display of the received content. Thus, DeBoor fails to disclose "a compiler generator having an input and an output, the input operative to receive an XML specification, the compiler generator operative to produce at the output a compiler" as recited in claim 16.

Thus, Applicant respectfully submits that for at least these reasons, claim 16 is patentably distinguishable over DeBoor. Therefore, Applicant respectfully requests that Examiner reconsider the rejection and withdraw it.

As claims 17 and 18 variously recite "a first interpreter," "a second interpreter" and "a compiler," the arguments advanced above with respect to claims 15 and 16 are hereby incorporated so as to apply to claims 17 and 18. Thus, Applicant respectfully submits that for at least these reasons, claims 17 and 18 are patentably distinguishable over DeBoor. Therefore, Applicant respectfully requests that Examiner reconsider the rejection and withdraw it.

Independent claim 19 recites, in part:

> setting the value of a global variable mod to refer to a module element describing the execution behavior of the root

As discussed above, DeBoor merely discloses an MMI which determines how content is displayed on a display screen where "the inclusion of predefined symbols in the content itself" indicates "instructions for the layout of the content on the page or screen." *See* DeBoor, col. 3, lines 59-66. Because the MMI in DeBoor merely determines the layout of content on a display screen, there is no "module element describing the execution behavior of the root" as recited in claim 19. Rather, DeBoor discloses using symbols in received content to format the received content for display without any execution (e.g., database query, mathematical operation) of the received content. Thus, DeBoor fails to disclose "a module element describing the execution behavior of the root" as recited in claim 19.

Thus, Applicant respectfully submits that for at least these reasons, claim 19 is patentably distinguishable over DeBoor. Therefore, Applicant respectfully requests that Examiner reconsider the rejection and withdraw it

As claim 20 is dependent from claim 19, all arguments advanced above with respect to claim 19 are hereby incorporated so as to apply to claim 20. Thus, Applicant respectfully submits that for at least these reasons, claim 20 is patentably distinguishable over DeBoor. Therefore, Applicant respectfully requests that Examiner reconsider the rejection and withdraw it.

Applicant has added new claims 22-27 for which Applicant requests consideration and examination. Applicant respectfully submits that these new claims are supported by the specification. Further, applicant notes that these claims correspond to allowed claims from the European Patent Office case corresponding to this application and are likewise believed to be patentable.

New independent claim 22 recites:

> A method for the direct execution of XML-documents comprising the steps of:
>
> defining a local behavior and process for each element of a XML-document;
>
> integrating executable instructions within at least one from the group of a document type definition (DTD), a XML document and a representation of the XML-document as a structure tree; and
>
> storing intermediate states by dynamically creating and redefining element attributes.

Hence, new claim 22 recites defining a local behavior and process for each element of an XML-document and integrating executable instructions with at least one XML-document or a document type definition (DTD). Integrating executable instructions with an XML-document or DTD definition allows the XML-document or DTD to execute a program, such as a mathematical computation, database query, data storage or other similar operation. Intermediate states caused by the executionExecution of the executable instructions causes intermediate states of the executable instructions are stored by dynamically creating and redefining element attribues. Hence, the element attributes of the XML-document are used to store states from the executable instructions. Thus, as the program defined by the XML-document is executed, the attribute elements of the XML-document itself are modified to reflect the execution.

DeBoor fails to disclose "storing intermediate states by dynamically creating and redefining element attributes" as recited in claim 1. In contrast, DeBoor discloses "(intermediate states of the process are stored and) replacing the tag with the second markup language page to form a combined markup language page (i.e. the elements are redefined)." DeBoor, col. 9, lines 9-10. This discloses the operation of a conventional markup language used by web-publishing systems for rendering the content of a web page for display on a wireless device. In describing the MMI, DeBoor explicitly states:

> Content handlers 114 are responsible for decoding the content data of a page corresponding to a fetched URL and displaying the content, or otherwise manipulating the content. A content handler 114 typically decodes content it receives and presents a page in the screen display 136, or portion thereof. Some content handlers 114 construct the page from data they receive from the memory 126 or over a communications link, while others display the state of the wireless communication device 100 or serve some other administrative function.

DeBoor, col. 12, lines 37-46. Thus, the MMI of DeBoor merely interpretes tags of a markup language to determine how to present data to a user. Replacing tags with markup pages to form a combined markup language page does not involve "storing intermediate states by dynamically creating and redefining element attributes," but merely displaying data according to a specified format. Thus, in DeBoor, there are no intermediate states to be stored, but merely a combined markup language page generated from replacing tags. Hence, DeBoor fails to disclose "storing intermediate states by dynamically creating and redefining element attributes," as recited in new claim 22.

New claims 23-27 depend from new claim 22 and variously recite additional patentable features, such as "integrating executable instructions by defining for each XML-element definition and its instances an action composed by simple executable actions, and actions which are references to either the action defined for one of the components of the element or to the action defined for any other element," "defining the composition of the action for at least one XML-element definition or instance by graphical flow charts representing sequential or concurrent control- and/or data-flow" and "representing system states in terms of n-dimensional data cubes and an open interface to the system by making the n-dimensional data cubes readable and writeable for other programming and/or database systems."

CONCLUSION

In sum, Applicant respectfully submits that claims 1-27, as presented herein, are patentably distinguishable over the cited references (including references cited, but not applied). Therefore, Applicant requests reconsideration of the basis for the rejections to these claims and requests allowance of them.

In addition, Applicant respectfully invites the Examiner to contact Applicant's representative at the number provided below if the Examiner believes it will help expedite furtherance of this application.

Respectfully Submitted,
PHILIPP W. KUTTER

Dated: June 18, 2007

By: /Brian G. Brannon/
Brian G. Brannon, Reg. No. 57,219
Attorney for Applicant
FENWICK & WEST LLP
801 California Street
Mountain View, CA 94041
Phone: (650) 335-7610
Fax: (650) 938-5200